

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Serial No. 09/870,614
Confirmation No. 1779

I hereby certify that this correspondence is being transmitted to the United States Patent & Trademark Office via electronic submission or facsimile on the date indicated below:

<u>11/15/2006</u>	<u>/Pamela Gerik/</u>
Date	Pamela Gerik

SUPPLEMENTAL APPEAL BRIEF

Sir:

Further to the Notice of Appeal faxed October 26, 2004 and received in the U.S. Patent and Trademark Office on the same day, Appellant presents this Supplemental Appeal Brief. This supplemental brief is filed in response to a Notice of Non-Compliant Brief mailed October 17, 2006. The Notice of Appeal was filed following mailing of a Final Office Action on August 13, 2004. Appellant hereby appeals to the Board of Patent Appeals and Interferences from a final rejection of claims 1-22 in the Final Office Action, and respectfully requests that this appeal be considered by the Board.

I. REAL PARTY IN INTEREST

The subject application is owned by International Business Machines Corporation, a corporation having its principal place of business at New Orchard Road, Armonk, New York, 10504, as evidenced by the assignment recorded at Reel 011888, Frame 0549.

II. RELATED APPEALS AND INTERFERENCES

Notices of Appeal have been filed for the following applications, which share a common specification with the application currently on appeal.

09/870,613: Notice of Appeal filed 2/07/05; Appeal Brief filed April 7, 2005.

09/870,615: Notice of Appeal filed 9/14/04; Appeal Brief filed November 9, 2004.

09/870,620: Notice of Appeal filed 12/7/04; Appeal Brief filed February 7, 2005.

09/870,621: Notice of Appeal filed 9/24/04; Appeal Brief filed November 23, 2004.

09/870,622: Notice of Appeal filed 8/24/04; Appeal Brief filed October 25, 2004.

09/870,624: Notice of Appeal filed 5/23/05; Appeal Brief filed July 25, 2005.

Application serial numbers 09/870,615, 09/870,622 and 09/870,624 share similar cited art references with the present application; however, dissimilar art is cited in the present application and application serial number 09/870,613, 09/870,620 and 09/870,621. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-22 are pending in the captioned case. Claims 1-22 stand rejected.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been filed subsequent to their final rejection. The Appendix hereto therefore reflects the current state of the claims.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellant's claimed invention relates to a display system (10, Fig. 1), a computer-readable storage device (18, Fig. 1) and a method (Fig. 15) for displaying a graphical representation (26, Fig. 2) of an object (24). In accordance with aspects of the present invention, however, the graphical representation of the object may, in some cases, be temporarily stored within a display buffer (e.g., memory 18 or processor 12 of Fig. 1) before the graphical representation is forwarded to a display device (16, Fig. 1). As described in more detail below, the Appellant's claimed invention more specifically relates to a system and method for enabling/disabling buffering of the graphical representation. (Specification – page 34, line 12 to page 35, line 25, and Abstract).

In some embodiments, the display system as recited in present claim 1 may include a display (16, Fig. 1), a display buffer (included within, e.g., memory 18 or processor 12) coupled to the display, and a processor (12). The processor may be adapted to execute an application program (APP 28, Fig. 2) which, when executed, produces images (26) upon the display. These so-called “images” may otherwise be referred to herein as graphical representations of an object (24). The object may be included within, e.g., a graphical user interface (GUI, Fig. 1) associated with the application program. (Specification – page 16, line 25 to page 17, line 15)

During a first mode, the images may be forwarded in sequence to the display. During a second mode, however, at least two of the images may be compiled as a combination image (e.g., combined images 228, 230, 232 and 234, Fig. 14) and presented to the display buffer before the combination image is forwarded to the display. (Specification – page 34, line 12 to page 35, line 10) In some cases, the application program may function to disable or enable buffering of the images by configuring the processor to execute in either the first or the second mode (see, Fig. 15). For example, the processor may be configured for executing in the first mode when the display is directly coupled to the processor, or in the second mode, when the display is remotely coupled to the processor. (Specification – page 35, lines 9-25)

In some embodiments, the computer-readable memory as recited in present claim 5 may include an operating system (OS 40, Fig. 2), an application program (APP 28) running on code compatible with the operating system, and a software component (API 22) invoked by the application program to display object code (24) which, when executed, produces a sequence of images (26) upon a display screen (16, Fig. 1). (Specification – page 16, line 25 to page 17, line 21 and page 18, lines 1-8) The software component can be configured during runtime of the application program to enable or disable buffering of the sequence of images as a combination image before the combination image is sent to the display screen (see, Fig. 15). (Specification – page 35, lines 9-25)

In some embodiments, the computer-readable storage device as recited in present claim 19 may include a windows-based operating system (OS 40, Fig. 2), an application program (APP 28) running under the operating system, an object (24) created at runtime by the application program, and an interface (API 22) independent of the operating system. (Specification – page 16, line 25 to page 17, line 21 and page 18, lines 1-8) The interface may be generally adapted for: (i) creating a graphical representation (26) of the object, and (ii) enabling or disabling buffering of the graphical representation of the object to a memory storage area (e.g., a display buffer) prior to displaying the graphical representation, as directed by the application program (see, Fig. 15). (Specification – page 35, lines 9-25)

In some embodiments, the method as recited in present claim 12 may include the steps of providing an application program (APP 28) running under an operating system (OS 40) (Specification – page 17, lines 17-21); creating a graphical representation of the object (e.g., images 26 of object 24) using an interface (API 22) independent of the operating system (Specification – page 18, lines 1-8), and enabling or disabling buffering of said graphical representation during runtime, as directed by the application program, and before the graphical representation is forwarded to a display device (16) (Specification – page 34, line 12 to page 35, line 25). As shown in Fig. 15, image buffering may be enabled if the display device is remotely coupled to the system running the application program, or disabled if the display device is directly coupled to the system.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 2, 5 and 6 stand rejected under 35 U.S.C. §102(b) as anticipated by U.S. Patent No. 4,951,229 to DiNicola et al. (hereinafter “DiNicola”).
2. Claims 3-4 and 7-10 stand rejected under 35 U.S.C. §103(a) over DiNicola in view of a web publication written by Amy Fowler entitled “*Mixing Heavy and Light Components*” (hereinafter “Fowler”).
3. Claims 11-13, 18 and 19 stand rejected under 35 U.S.C. §103(a) over DiNicola in view of a web publication written by Sun Microsystems entitled “*Introducing Swing*” (hereinafter “Sun”).
4. Claims 14-17 stand rejected under 35 U.S.C. §103(a) over DiNicola, Fowler and Sun.
5. Claims 20-22 stand rejected under 35 U.S.C. §103(a) over DiNicola.

VII. ARGUMENT

The contentions of the Appellant with respect to the ground of rejection presented for review, and the basis thereof, with citations of the statutes, regulations, authorities, and parts of the record relied upon are presented herein for consideration by the Board. Details as to why the rejections cannot be sustained are set forth below.

A. Patentability of claims 1, 2, 5, and 6 under 35 U.S.C § 102(b)

Claims 1, 2, 5 and 6 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 4,951,229 to DiNicola et al. (hereinafter “DiNicola”). As described in more detail below, the §102(b) rejection of claims 1, 2, 5 and 6 is hereby traversed.

The standard for “anticipation” is one of fairly strict identity. A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art of reference. *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); MPEP 2131. Using this standard, Applicants submit the cited art fails to disclose each and every element of the currently pending claims, some distinctive features of which are set forth in more detail below.

1. Patentability of claims 1 and 2 under 35 U.S.C § 102(b)

DiNicola fails to anticipate a display system including a processor, which is adapted to produce images upon a display, wherein during a second mode, at least two of the images are compiled as a combination image and presented to a buffer before they are forwarded to the display, as recited in present claim 1. Independent claim 1 states in part:

A display system, comprising: a display; a display buffer ... and a processor adapted to execute an application program which, when executed, produces images upon the display ... wherein during a second mode the images are compiled as a combination image of at least one of said images drawn over at least another of said images and presented to the buffer before being forwarded to the display.

The invention as described in present claim 1 provides a display system that improves the performance of conventional Swing-based APIs. For example, conventional Swing-based APIs default to buffering their graphical output. In other words, instead of directly displaying graphics content, Swing-based APIs are required to draw the graphical content to an image buffer, and then display the entire buffer *en masse*. See, e.g., Specification, page 34, lines 12-15.

Fig. 14 of the present specification illustrates how conventional Swing-based APIs default to buffering their graphical output (e.g., Frame 220, Panel 222, Buttons 224 and 226). For example, when a Frame (220) and its contents (222, 224 and 226) are to be displayed, they are first drawn to an image buffer (e.g., a memory buffer) where the images of the objects (e.g., the Frame, Panel and Buttons) are combined into a composite image. However, once the images are combined, bitmaps of overlying components (e.g., Buttons 224 and 226) may sometimes

replace portions of the bitmap of an underlying component (e.g., certain portions of panel 222). *See*, e.g., Specification, page 34, lines 12-26 and Fig. 14.

In some cases, the default buffering of Swing images may lead to sub-optimal performance. As stated in the specification, for example, “if several Swing objects are present within multiple levels of containment in a layout, the overhead associated with buffering may defeat the hardware acceleration capability of video graphics hardware within the computer. This would then result in a slower-than-necessary display of the objects. On the other hand, buffering may be advantageous in a system involving remote displays. Rather than sending the graphic representation for each object in the layout to a remote terminal in the sequence in which the objects are drawn, it may be faster in such cases to draw the entire layout to a local buffer, and then send the buffered contents in a single transmission, thereby minimizing network traffic and bandwidth.” *See*, Specification, page 34, line 28 – page 35, line 7.

The invention as recited in claim 1 improves the performance of conventional Swing-based APIs by providing a display system, which is uniquely configured for enabling/disabling image buffering before the images are sent to the display. If buffering is disabled, the images are forwarded in sequence to the display (e.g., during a first mode). If buffering is enabled, at least two of the images may be compiled as a combination image. The combination image may then be presented to a display buffer before it is forwarded to the display (e.g., during a second mode). In some cases, the determination of whether to enable or disable buffering may depend on whether a display device is locally or remotely coupled to the display system. For example, buffering may be disabled when the display device is directly coupled to the display system and enabled when the display device is remotely coupled to the display system. *See*, e.g., Specification, page 35, lines 9-25; Fig. 15.

DiNicola discloses an apparatus and method for managing multiple images in a graphic display system (DiNicola, Title). DiNicola, however, fails to anticipate a display system as specifically recited in present claim 1. In particular, DiNicola fails to provide teaching or suggestion for a display system including a display, a display buffer and a processor, which is adapted to produce images upon the display, such that during a second mode, at least two of the

images are compiled as a combination image and presented to the display buffer before the combination image is forwarded to the display.

On page 2 of the Office Action mailed January 29, 2004, the Examiner suggests, “DiNicola teaches, in column 2, line 59 through column 3, line 16, especially column 3, lines 10-13, a system that can be configured to ... combine two or more of the images and send them as a composite display image.” The Examiner further suggests that DiNicola provides teaching for “presenting the image to the buffer before forwarding ... in column 3, lines 62-68 ...” (Office Action, page 3). The Appellants assert that, although DiNicola may “combine two or more of the images to form a composite display image” (DiNicola -- column 3, lines 10-13), DiNicola does not teach or suggest that the composite display image may be presented to a display buffer before it is forwarded to the display, as taught in present claim 1.

In the passage cited by the Examiner (e.g., DiNicola -- column 3, lines 62-68), DiNicola describes how output messages from processor 10 may be formatted and stored in memory buffers 24, 26, 28 and 30, as shown in Fig. 1 of DiNicola. The formatted output messages of DiNicola, however, are not equivalent to the presently claimed “combination image,” since DiNicola’s image mixing process is performed downstream of memory buffers 24, 26, 28 and 30. For example, DiNicola states, “[i]mage mixer 32 reads the encoded graphics data from the memory buffers 24, 26, 28 and 30 and performs the necessary decoding and image mixing ... The merged images are passed through the color translation table 34 which generates the appropriate control signals to be passed on data line 40 to display monitor 50.” (DiNicola -- column 4, lines 14-30; Fig. 1).

From the cited passage, DiNicola clearly states that the combined images (i.e., the mixed images) are passed directly to display monitor 50. DiNicola does not teach or suggest that the combined images may be stored within memory buffers 24, 26, 28 and 30, or within any other memory buffer, before the combined images are forwarded to the display monitor. As such, DiNicola fails to provide teaching or suggestion for the presently claimed display system, which when operating in a second mode, requires at least two images to be compiled into a composite image, which is presented to a display buffer before the composite image is forwarded to the

display. As a consequence, the teachings of DiNicola cannot be relied upon to anticipate all limitations of present claim 1.

2. Patentability of claims 5 and 6 under 35 U.S.C § 102(b)

DiNicola fails to disclose a computer-readable memory including a software component, which can be configured during runtime of an application program to enable or disable buffering of a sequence of images as a combination image before sending the combination image to a display. Independent claim 5 recites, in part:

A computer-readable memory, comprising... an application program running on code compatible with the operating system; and a software component invoked by the application program to display object code which, when executed, produces a sequence of images upon a display screen, wherein the software component can be configured during runtime of the application program to enable or disable buffering of the sequence of images as a combination image before sending the combination image to the display.

As noted above, the presently claimed case improves upon conventional Swing-based APIs by providing a means for enabling or disabling the buffering of graphical content. As set forth in claim 5, the presently claimed case provides a software component, which can be configured during runtime of an application program to enable or disable buffering of a sequence of images as a combination image before sending the combination image to a display. As described in more detail below, DiNicola fails to provide teaching or suggestion for the presently claimed software component, and thus, cannot be relied upon to anticipate all limitations of present claim 5.

With regard to claim 5, the Examiner suggests that DiNicola provides teaching for a software component, which can be configured during runtime of an application program to enable or disable buffering of a sequence of images as a combination image before sending the combination image to a display. For example, the Examiner states that DiNicola provides teaching for “optional buffering of the sequence of images as a combination image before sending the combination image to the display” by disclosing (e.g., DiNicola -- column 3, lines

62-68; column 5, lines 18-32) “an intermediate buffer that is not required, as a matter of efficiency, but mentioned in the reference.” (Final Office Action, page 4). However, and as described in more detail below, DiNicola does not include an intermediate buffer within his graphical display system, nor does he provide desirability for the Examiner’s so-called “optional” buffering.

In order to provide “optional buffering of a sequence of images as a combination image before sending the combination image to the display,” the graphical display system of DiNicola would have to include an “intermediate buffer” between the mixing stage and the display monitor. As shown in Fig. 1, however, an intermediate buffer simply does not exist between the image mixer (32) and display monitor (50) included within the graphical display system of DiNicola.

Furthermore, and contrary to the Examiner’s suggestions, DiNicola does not teach or suggest that the intermediate buffer may be an “optional” feature of the invention. Instead, DiNicola describes the intentional absence of an intermediate frame buffer as being a distinct advantage over prior art display systems. For example, DiNicola states:

This invention ... provides an inventive capability to combine images from separate lateral bit encoded bit planes. Image mixing is performed by image mixer 32 which reads data from the bit planes, combines the image data and transmits it directly to the display monitor. There is no requirement for an intermediate frame buffer to store the results of the image mixing operation. This is a significant advantage because a change to one of the bit planes does not require the complete regeneration of an intermediate frame buffer.” (DiNicola, column 5, lines 18-27, emphasis added).

By specifically stating that there is no requirement for an intermediate frame buffer, DiNicola teaches away from using an intermediate frame buffer and provides absolutely no teaching, suggestion or motivation for possibly including the intermediate frame buffer as an “optional” feature.

As noted in section 2143.02 of the MPEP, a prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983). Appellants assert that a prior art reference cannot be used to anticipate a claimed limitation if that reference specifically teaches away from the limitation.

For at least the reasons set forth above, Appellants assert that DiNicola fails to anticipate all limitations of present claims 1 and 5. Therefore, Appellants assert that independent claims 1, 5, and all claims dependent therefrom, are not anticipated by the cited art. Appellants further assert that an anticipatory rejection of the current claims cannot be sustained and, therefore, request that this rejection be reversed.

B. Patentability of claims 3-4 and 7-10 under 35 U.S.C § 103(a)

Claims 3-4 and 7-10 were rejected under 35 U.S.C. §103(a) over DiNicola in view of a web publication written by Amy Fowler entitled “*Mixing Heavy and Light Components*” (hereinafter “Fowler”). Because claims 3-4 and 7-10 are dependent from independent claims 1 and 5, respectfully, the arguments presented above for patentability of claims 1 and 5 apply equally to claims 3-4 and 7-10, and are herein incorporated by reference. In addition to the 35 U.S.C. §102 arguments presented above with respect to claims 1 and 5, arguments are provided below to establish patentability of the current claims under 35 U.S.C. §103(a). As such, the §103(a) rejection of claims 3-4 and 7-10 is hereby traversed.

MPEP 2143 establishes the basic requirements in finding a *prima facie* case of obviousness. As described, to establish a case of *prima facie* obviousness of a claimed invention, three criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the references or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art references must teach or suggest all the claim limitations. Specifically, “all words in a claim must be considered when judging the patentability of that claim against the prior art.” *In re Wilson* 424 F.2d. 1382, 1385 (CCPA

1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Fowler cannot be combined with DiNicola to provide teaching, suggestion or motivation for the display system, as recited in present claim 1, or the computer-readable memory, as recited in present claim 5. The display system as recited in present claim 1 includes a processor, which is adapted to produce images upon a display, such that during a second mode, at least two of the images are compiled as a combination image and presented to a buffer before the combination image is forwarded to the display. In a similar manner, the computer-readable memory as recited in present claim 5 includes a software component, which can be configured during runtime of an application program to enable or disable buffering of a sequence of images as a combination image before the combination image is sent to a display.

As noted above, DiNicola simply fails to provide teaching or suggestion for the display system and computer-readable memory, as set forth in present claims 1 and 5. More specifically, DiNicola fails to provide teaching or suggestion for combining at least two images as a combination image and presenting the combination image to a buffer, before the combination image is forwarded to a display (as recited in claim 1). Furthermore, DiNicola fails to provide even the slightest desirability for enabling or disabling the buffering of a sequence of images as a combination image before the combination image is sent to a display (as recited in claim 5). As described in more detail below, DiNicola cannot be modified or combined with the teachings of Fowler to provide teaching for the limitations forth in present claims 1 and 5.

As noted above in the §102 arguments, the Examiner suggests that DiNicola provides teaching for “optional buffering of the sequence of images as a combination image before sending the combination image to the display” by disclosing (e.g., in column 3, lines 62-68 and column 5, lines 18-32) “an intermediate buffer that is not required, as a matter of efficiency, but mentioned in the reference.” (Final Office Action, page 4). The Appellants have already shown that the graphical display system of DiNicola does not include an intermediate buffer (see, Fig. 1 of DiNicola). As described in more detail below, DiNicola lacks the necessary motivation that

would enable one skilled in the art to modify the display system of DiNicola to include the intermediate buffer proposed by the Examiner.

The Examiner assumes the position that, since an intermediate buffer is mentioned in the reference, it can be included as an “optional feature.” However, the mere mention of an intermediate frame buffer within the cited art reference would not motivate one skilled in the art to modify the graphical display system of DiNicola to include such a buffer, since DiNicola specifically teaches away from doing so. For example, and as noted above, DiNicola describes the intentional absence of an intermediate frame buffer as a distinct advantage over prior art display systems. By teaching away from using an intermediate frame buffer, DiNicola provides absolutely no motivation that would enable one skilled in the art to include such a buffer as an “optional feature” of the invention. In other words, Appellants assert that not only does DiNicola fail to anticipate all limitations of present claims 1 and 5, DiNicola cannot be modified to do so.

In addition, the teachings of Fowler cannot be combined with DiNicola to overcome the deficiencies therein. Appellants realize that Fowler is not relied upon for teaching the limitations of present claims 1 and 5, but instead, is relied upon for mentioning certain AWT and Swing components (e.g., frames, panels and buttons). However, since currently rejected claims 3-4 and 7-10 are dependent on claims 1 and 5 and are, therefore, patentable for at least the same reasons as those claims, Appellants wish to address the lack of teaching within Fowler for the limitations of independent claims 1 and 5.

In a publication entitled *Mixing Heavy and Light Components*, Fowler provides a brief overview of certain problems that may be encountered when mixing heavyweight (e.g., AWT) components and lightweight (e.g., Swing) components, and presents certain guidelines that should be followed to circumvent such problems. However, Fowler fails to mention that images generated by Swing components (e.g., images of frames, panels and buttons) could be buffered. Since Fowler fails to mention a display buffer, buffering of images, or any means or desirability for enabling/disabling a display buffer, Fowler cannot be relied upon to provide teaching or suggestion for the limitations recited in present claims 1 and 5.

In addition to explicit lack of teaching, Fowler lacks the necessary motivation that would enable one skilled in the art to modify the teachings of Fowler to provide the patentable features recited in claims 1 and 5. For example, Fowler fails to mention that Swing-generated images are automatically buffered before the images are sent to a display device. In addition, although Fowler mentions a few of the problems encountered when mixing Swing and AWT components, Fowler does not mention the potential problems associated with default buffering of Swing-generated images. For example, Fowler does not suggest that buffering may unnecessarily slow down the display of images, when those images are to be displayed on a local display device. Therefore, Fowler provides absolutely no motivation for providing the capability to enable and disable the buffering of images, and as a consequence, cannot be modified to do so.

Since DiNicola and Fowler each fail to teach, suggest or provide motivation for enabling and disabling the buffering of images, DiNicola and Fowler cannot be combined to do so. Even if the proposed combination were made (without sufficient motivation to do so), Appellants assert that the combined teachings of DiNicola and Fowler would still fail to disclose all limitations of present claims 1 and 5.

C. Patentability of claims 11-13 and 18-19 under 35 U.S.C § 103(a)

Claims 11-13 and 18-19 were rejected under 35 U.S.C. §103(a) over DiNicola in view of a web publication written by Sun Microsystems entitled “*Introducing Swing*” (hereinafter “Sun”). Because claims 11 is dependent from independent claim 5, the arguments presented above for patentability of claim 5 over the teachings of DiNicola and Fowler apply equally to claim 11, and are herein incorporated by reference. Additional arguments are provided below to establish patentability of independent claims 5, 12, and 19 over the teachings of DiNicola and Sun. Because claims 13 and 18 depend from independent claim 12, the arguments presented below for the patentability of claim 12 apply equally to claims 13 and 18. As such, the §103(a) rejection of claims 11-13 and 18-19 is hereby traversed.

MPEP 2143 establishes the basic requirements in finding a *prima facie* case of obviousness. As described, to establish a case of *prima facie* obviousness of a claimed invention, three criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the references or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art references must teach or suggest all the claim limitations. Specifically, “all words in a claim must be considered when judging the patentability of that claim against the prior art.” *In re Wilson* 424 F.2d 1382, 1385 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Sun cannot be combined with DiNicola to provide teaching, suggestion or motivation for a software component, which is configured during runtime of an application program to enable or disable buffering of a sequence of images as a combination image before the combination image is sent to a display, as recited in claim 5. As noted above, DiNicola simply fails to provide teaching, suggestion, or motivation for the software component recited in present claim 5.

Appellants realize that Sun is not relied upon for teaching the limitations of present claim 5, but instead, is relied upon for describing how Swing-based application programs can be run on different operating systems (such as Windows and Unix). However, since currently rejected claim 11 is dependent on claim 5 and is, therefore, patentable for at least the same reasons as claim 5, Appellants wish to address the lack of teaching within Sun for the limitations of independent claim 5.

The article entitled *Introducing Swing* does just as the title implies – it briefly outlines the Swing architecture and the basic differences between Swing and AWT. However, *Introducing Swing* (otherwise referred to as Sun) fails to provide teaching or suggestion for the software component recited in present claim 5, which is configured to enable or disable buffering of a sequence of images as a combination image before the combination image is sent to a display. In addition, Sun fails to mention the use of display buffers, buffering of images, the manner in

which Swing-generated images are automatically buffered by default, or the potential problems associated with default buffering. As a consequence, Sun cannot be relied upon to provide teaching or suggestion for a software component specifically configured to enable or disable buffering of a sequence of images as a combination image before the combination image is sent to a display.

In addition to explicit lack of teaching, Sun lacks the necessary motivation that would enable one skilled in the art to modify the teachings of Sun to provide a software component, which allows a sequence of images to be compiled as a combination image and presented to a display buffer before the combination image is forwarded to a display device. Therefore, and as described in more detail below, Sun cannot be modified or combined with DiNicola to overcome the deficiencies therein. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed.Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992); MPEP 2143.01.

As noted above, Sun provides absolutely no teaching or suggestion for a display buffer, buffering of images, or any means (e.g., a software component) for enabling/disabling a display buffer. In addition, and contrary to the presently claimed case, Sun does not disclose the manner in which Swing-generated images are automatically buffered by default, or the potential problems associated with default buffering. Therefore, Sun provides absolutely no motivation that would enable one skilled in the art to modify the teachings of Sun to include a software component, which is capable of enabling/disabling the buffering of a sequence of images as a combination image before the combination image is sent to a display, as recited in present claim 5. As a consequence, Sun cannot be modified to provide teaching or suggestion for the limitations of present claim 5.

However, statements in the Office Action mailed January 29, 2004 and in the final Office Action mailed April 29, 2004 suggest that “DiNicola teaches the system that either transmits images to the display sequentially or as a combination image ... [and] Sun teaches Swing[,] which is stated in the specification of the application to default to buffering it’s output, similar to

that of DiNicola ...” (Office Action, page 8; Final Office Action, page 9). As such, the Examiner appears to rely on the Applicant’s own disclosure to provide motivation for combining the teachings of DiNicola and Sun. This simply cannot be done. As noted in section 2143 of the MPEP, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant’s disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

DiNicola and Sun each fail to teach, suggest or provide motivation for enabling and disabling the buffering of images. In addition, DiNicola and Sun cannot be modified or combined to provide such teaching, since neither DiNicola nor Sun suggest even the slightest desirability for making the proposed combination or modification. Though the Applicant’s disclosure mentions Swing’s default buffering capability (which may be considered an inherent aspect of Swing), the potential problems associated therewith and the particular solution provided by the presently claimed case would not be apparent to one skilled in the art given the teachings of Sun and DiNicola. In other words, the Applicant’s own disclosure cannot be used as motivation to combine or modify the prior art references when motivation cannot be found in the prior art references themselves.

DiNicola and Sun each fail to disclose a computer-readable storage device (claim 19) and method (claim 12) for displaying an object by enabling or disabling buffering of a graphical representation of the object to a memory storage area prior to displaying the graphical representation. Independent claim 12 recites in part:

A method for displaying an object, comprising ... creating a graphical representation of the object using an interface independent of the operating system; and enabling or disabling buffering of said graphical representation during runtime as directed by the application program.

Independent claim 19 recites in part:

A computer-readable storage device, comprising ... an interface independent of the operating system, and adapted for: creating a graphical representation of the object; and enabling or disabling buffering of the graphical representation of the object to a memory storage area prior to displaying the graphical representation, as directed by the application program.

As described in the section 102 and 103 arguments presented above, DiNicola simply fails to provide teaching, suggestion, or motivation for the limitations of claim 5, which relate to enabling and disabling the buffering of a sequence of images as a combination image before the combination image is forwarded to a display.

According to a more general embodiment of the invention, present claims 12 and 19 each recite limitations on enabling/disabling the buffering of a “graphical representation of an object.” This is also not taught or suggested by DiNicola. For example, the only instance in which DiNicola actually utilizes image buffering occurs when “[o]utput messages from the processor 10 are formatted and stored in memory buffers, or bit planes 24, 26, 28, and 30.” (DiNicola -- column 3, lines 62-66). However, DiNicola does not teach or suggest that memory buffers 24, 26, 28 and 30 could be disabled in certain circumstances. Since no other memory buffers exist within the graphical display system of DiNicola (*see* above arguments as to why the “intermediate frame buffer” of DiNicola does not and cannot be used to provide “optional buffering,” as suggested by the Examiner), DiNicola cannot be relied upon to provide teaching or suggestion for the presently claimed limitations of enabling/disabling the buffering of a graphical representation of an object.

In addition, DiNicola fails to provide motivation for the presently claimed computer-readable storage device and method by failing to even suggest that memory buffers 24, 26, 28 and 30 could be disabled in certain circumstances. In other words, there is simply no motivation within DiNicola that would enable one skilled in the art to modify the teachings of DiNicola, such that the capability exists for enabling and disabling memory buffers 24, 26, 28, and 30.

Furthermore, if the teachings of DiNicola were somehow modified (without sufficient motivation to do so) to provide capability for enabling/disabling memory buffers 24, 26, 28 and 30, Appellants contend that the modified invention of DiNicola would be rendered unsatisfactory for its intended purpose. As noted in section 2143.01 of the MPEP, if the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

For example, the intended purpose of DiNicola is to support both bit encoded and multiplane lateral bit encoding techniques. (See, DiNicola -- column 2, lines 49-58). DiNicola states that, in bit plane encoding, “[t]he information stored in any one plane represents only a subset of the information required to create the picture element on the display monitor. None of the planes represent the full structure of the image; it is only the combination of the several planes that allows the final image to be made apparent. Data is read from each bit plane simultaneously from the same relative bit location. For example, bits c_0 , c_1 and c_2 would be read from the bit planes to form a single picture element for display.” (DiNicola -- column 4, lines 39-48). Therefore, in order to achieve bit plane encoding, DiNicola states that data must be stored in the bit planes (i.e., memory buffers 24, 26, 28, and 30) before individual bits can be “mixed” to form a single picture element for display. Since bit plane encoding would not be possible if memory buffers 24, 26, 28, and 30 were disabled, DiNicola cannot be modified to include the capability for enabling and disabling the memory buffers, since such modification would not allow the graphical display system of DiNicola to support both bit encoded and multiplane lateral bit encoding techniques.

In addition, Sun cannot be modified or combined with DiNicola to overcome the deficiencies therein. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed.Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992); MPEP 2143.01.

As noted above, Sun provides absolutely no teaching or suggestion for a display buffer, buffering of images, or any means for enabling/disabling image buffering. Though Sun provides a brief overview of Swing, Sun does not disclose the manner in which Swing-generated images are automatically buffered by default, or the potential problems associated with default buffering. Therefore, Sun provides no motivation that would enable one skilled in the art to modify the teachings of Sun to include a computer-readable storage device or method for enabling/disabling the buffering of a graphical representation of an object, as recited in present claims 12 and 19. As a consequence, Sun cannot be modified to provide teaching or suggestion for the aforementioned limitation of present claims 12 and 19.

Since DiNicola and Sun each fail to teach, suggest or provide motivation for enabling and disabling the buffering of images (i.e., graphical representations of objects), DiNicola and Sun cannot be combined to do so. Even if the proposed combination were made (without sufficient motivation to do so), Appellants assert that the combined teachings of DiNicola and Sun would still fail to disclose all limitations of present claims 12 and 19.

D. Patentability of claims 14-17 under 35 U.S.C § 103(a)

Claims 14-17 were rejected under 35 U.S.C. §103(a) over DiNicola, Fowler and Sun. Because claims 14-17 are dependent from independent claim 12, the arguments presented above for patentability of claim 12 over the teachings of DiNicola and Sun apply equally to claims 14-17, and are herein incorporated by reference. Additional arguments are provided below to establish patentability of the independent claim 12 over the teachings of Fowler. Because claims 14-17 depend from independent claim 12, the arguments presented below for patentability of claim 12 apply equally to claims 14-17. As such, the § 103(a) rejection of claims 14-17 is hereby traversed.

DiNicola, Fowler and Sun each fail to disclose a method for displaying an object by enabling or disabling buffering of a graphical representation of the object, and furthermore, cannot be combined or modified to do so. As noted above for the patentability of claims 12 and 19, DiNicola and Sun each fail to teach, suggest or provide motivation for enabling/disabling the buffering of images (i.e., a graphical representation of an object), and more specifically, cannot be combined or modified to do so.

As noted above for the patentability of claims 3-4 and 7-10, Fowler fails to teach, suggest or provide motivation for enabling and disabling the buffering of images. In particular, Fowler fails to disclose a display buffer, buffering of images, or any means for enabling/disabling the buffering of images, regardless of whether those images form a “combination image,” as recited in claims 1 and 5, or merely a “graphical representation of an object”, as recited in claim 12. Therefore, Fowler cannot be relied upon to disclose a method for displaying an object, where the method includes enabling or disabling the buffering of a graphical representation of an object.

As such, Fowler fails to provide teaching, suggestion or motivation for all limitations of claim 12.

The only reference that even mentions image buffering is DiNicola. However, and as repeatedly stated above, DiNicola does not teach the capability for enabling/disabling the buffering of images, nor can the teachings of DiNicola be modified to do so. Since Fowler and Sun fail to mention buffering, Fowler and Sun cannot be modified or combined with DiNicola to overcome the deficiencies therein.

E. Patentability of claims 20-22 under 35 U.S.C § 103(a)

Claims 20-22 were rejected under 35 U.S.C. §103(a) over DiNicola. Because claims 20-21 and 22 are dependent from claims 1 and 19, respectively, the arguments presented above for patentability of claims 1 and 19 apply equally to claims 20-22, and are herein incorporated by reference. As such, the §103(a) rejection of claims 20-22 is hereby traversed.

As noted in the arguments present above for the patentability of claims 1 and 19, DiNicola fails to teach, suggest or provide motivation for enabling/disabling the buffering of images. The lack of teaching within DiNicola applies to images that form a “combination image”, as recited in claim 1, or merely a “graphical representation of an object,” as recited in claim 19. In addition, the above arguments conclusively show that the teachings of DiNicola cannot be combined or modified to provide such teaching.

For the foregoing reasons, Appellant asserts that independent claims 1, 5, 12 and 19, as well as claims dependent therefrom, are patentably distinct over DiNicola, Fowler and Sun. Contrary to the characterizations made in the various Office Actions, the cited references cannot be properly combined or modified to provide teaching for all limitations recited in claims 1, 5, 12 and 19. Accordingly, Appellants assert that a *prima facie* case of obviousness has not been duly set out and, therefore, request that this rejection be reversed.

* * *

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-21 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the required fees to International Business Machines deposit account number 09-0447.

Respectfully submitted,
/Kevin L. Daffer/
Kevin L. Daffer
Reg. No. 34,146
Attorney for Appellant

Customer No. 35617
Date: November 15, 2006
JMF

VIII. CLAIMS APPENDIX

The present claims on appeal are as follows.

1. A display system, comprising:

a display;

a display buffer coupled to the display; and

a processor adapted to execute an application program which, when executed, produces images upon the display, wherein during a first mode the images are forwarded in sequence to the display, and wherein during a second mode the images are compiled as a combination image of at least one of said image drawn over at least another of said images and presented to the buffer before being forwarded to the display.

2. The display system as recited in claim 1, wherein the application program disables or enables buffering of the images by configuring the processor to execute in either the first or second mode.

3. The display system as recited in claim 1, wherein the images comprise frame, panel and button images.

4. The display system as recited in claim 1, wherein the application program comprises a Java program.

5. A computer-readable memory, comprising:

an operating system,

an application program running on code compatible with the operating system; and

a software component invoked by the application program to display object code which,
when executed, produces a sequence of images upon a display screen, wherein the
software component can be configured during runtime of the application program
to enable or disable buffering of the sequence of images as a combination image
before sending the combination image to the display.
6. The memory as recited in claim 5, wherein the object code is part of a graphical user
interface associated with the application program.
7. The memory as recited in claim 5, wherein the software component comprises an
application program interface of code which translates between code within the application
program and the operating system.
8. The memory as recited in claim 7, wherein a behavior of the application program
interface emulates that of a second application program interface based on windowing protocols
of a windows-based version of said operating system.
9. The memory as recited in claim 8, wherein the second application program interface
comprises a Java abstract windowing toolkit.
10. The memory as recited in claim 5, wherein the application program is written in Java
programming language.
11. The memory as recited in claim 5, wherein the operating system comprises a Windows,
Unix or OS/2 computer operating system.

12. A method for displaying an object, comprising:
- providing an application program running under an operating system;
- creating a graphical representation of the object using an interface independent of the operating system; and
- enabling or disabling buffering of said graphical representation during runtime as directed by the application program.
13. The method as recited in claim 12, wherein said creating comprises compiling the object as code that includes part of a graphical user interface associated with the application program.
14. The method as recited in claim 12, wherein said creating comprises implementing a call routine to compile a software component that includes an application program interface between the application program and the operating system.
15. The method as recited in claim 14, wherein a behavior of the application program interface emulates that of a second application program interface based on the operating system.
16. The method as recited in claim 15, wherein the second application program interface comprises a Java abstract windowing toolkit.
17. The method as recited in claim 12, wherein the application program is written in Java programming language.
18. The method as recited in claim 12, wherein the operating system comprises a Windows, Unix or OS/2 computer operating system.
19. A computer-readable storage device, comprising:
- a windows-based operating system;
- an application program running under the operating system;

an object created at runtime by the application program;

an interface independent of the operating system, and adapted for:

creating a graphical representation of the object; and

enabling or disabling buffering of the graphical representation of the object to a memory storage area prior to displaying the graphical representation, as directed by the application program.

20. The display system as recited in claim 2, wherein the processor executes in the first mode when the display is directly coupled to the processor.

21. The display system as recited in claim 2, wherein the processor executes in the second mode when the display is remotely coupled to the processor.

22. The computer-readable storage device as recited in claim 19, further comprising a software component associated with the interface and adapted to: (i) determine whether the application program is operating in a remote or direct mode, and (ii) create a peer component to enable or disable buffering of the graphical representation of the object based on the determination made by the software component.

IX. EVIDENCE APPENDIX

No evidence has been entered during the prosecution of the captioned case.

X. RELATED PROCEEDINGS APPENDIX

Notices of Appeal have been filed for the following applications, which share a common specification with the application currently on appeal.

09/870,613: Notice of Appeal filed 2/07/05; Appeal Brief filed April 7, 2005.

09/870,615: Notice of Appeal filed 9/14/04; Appeal Brief filed November 9, 2004.

09/870,620: Notice of Appeal filed 12/7/04; Appeal Brief filed February 7, 2005.

09/870,621: Notice of Appeal filed 9/24/04; Appeal Brief filed November 23, 2004.

09/870,622: Notice of Appeal filed 8/24/04; Appeal Brief filed October 25, 2004.

09/870,624: Notice of Appeal filed 5/23/05; Appeal Brief filed July 25, 2005.

Application serial numbers 09/870,615; 09/870,622; and 09/870,624 share similar cited art references with the present application; however, dissimilar art is cited in the present application and application serial number 09/870,613; 09/870,620; and 09/870,621. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.